Docket No. 50325-0552 (Seq. No. 4434)

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of

Cassio GOLDSCHMIDT	Confirmation No.: 6357
Serial No.: 09/905,306	Group Art Unit: 2672
Filed: July 13, 2001	Examiner: Motilewa GOOD-JOHNSON

Title: INCREMENTAL PLOTTING OF NETWORK TOPOLOGIES AND OTHER GRAPHS THROUGH USE OF MARKUP LANGUAGE

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

DECLARATION UNDER 37 C.F.R. 1.131

I, CASSIO BRUN GOLDSCHMIDT, pursuant to 37 C.F.R. § 1.131, declare:

1. I am the sole inventor named in the above-referenced application.

2. I understand that the Office Action mailed January 27, 2005 rejects the pending claims of this application by relying in part on U.S. Patent Application Publication No. 2002/0158897 A1 (BESAW et al., hereinafter "BESAW"), filed on April 30, 2001 and published on October 31, 2002.

3. I make this declaration for the purpose of establishing that a functioning version of a computer program application that embodies an invention disclosed and claimed in the above-referenced application (hereinafter "the subject invention") was created and completed in the United States at a date prior to April 30, 2001, which is the filing date of BESAW.

4. I conceived of, created and completed a functioning implementation of the subject invention, embodied in a computer program application alternatively referred to in attached exhibits as the "TopoML", the "AD topology application" and the "AD

application" (all referred to herein as the "TopoML application"), long prior to April 30, 2001. At that time I was employed as a Softwarc Engineer III by Cisco Systems, Inc., the parent corporation of Cisco Technology, Inc., the assignee of the present application.

5. I developed the "TopoML" programming language, which is a programming tool that was designed to be used to automatically layout and incrementally plot network topologies and other graphs. As established by the attached documentary exhibits, "TopoML" was a functioning programming language, and using that language, I completed a functioning version of a computer program application (the TopoML application) that embodies the subject invention, long prior to April 30, 2001.

6. Structurally, the TopoML application comprises a CGI program and an applet program, both of which function in compliance with the TopoML language, and which execute in conjunction with a layout server to perform the actions recited in Claims 1, 4-25, and 35 of the present application. The TopoML application functions as follows. The CGI program operates, in response to a request for a graphic topologic display, to retrieve from a data source and to convert topology information to a TopoML markup language document that conforms to a TopoML document type definition or schema, attached hereto as Exhibit 2 and Exhibit 3, respectively. The TopoML document is provided as input to an applet program, which operates to read and interpret the TopoML document in accordance with the TopoML document type definition or schema, and to provide directions to a layout server, which operates to plot a graphical representation of the topology information. This graphical representation is displayed, such as by a browser, under the direction of the applet program. Certain user interactions with the graphical display may cause the applet program to direct the browser to send another request to the CGI program, which in turn retrieves additional topology information and converts this additional topology information to a TopoML-conforming markup language document, on which the applet program and layout server operate as previously described, without repeating the same operations on the original markup language document. The applet program was constructed based on the programming constructs attached hereto as Exhibit 4 and Exhibit 5.

7. Attached, as Exhibit 1, is a true and correct redacted copy of a document entitled "TopoML". This document describes a purpose and functionality of the

"TopoML" language. This document refers to a functioning computer program application, the TopoML application, that embodies the subject invention described and claimed in the above-referenced application and which I implemented with the "TopoML" language. This document includes an associated screenshot generated by the TopoML application. This document refers to programming elements of the "TopoML" language. One programming element referred to in this document is "TopoML DTD", attached hereto as Exhibit 2. Another programming element referred to in this document is "W3CSchema", attached hereto as Exhibit 3. The date of the "TopoML" document is not shown on the document, but the actual date is long prior to April 30, 2001. The "TopoML" document is hereby submitted as probative of a reduction to practice of the subject invention prior to April 30, 2001.

8. Attached, as Exhibit 2, is a true and correct redacted copy of the contents of a document entitled "TopoML.dtd", which is referred to as "TopoML DTD" in the document of Exhibit 1. This document is an XML Document Type Definition (DTD), which describes the elements in a "TopoML" document and represents the interrelationship between the attributes and elements of a "TopoML" document. The CGI program of the TopoML application (referred to in Section 6 of this affidavit) operates to retrieve from a data source and to convert topology information to a TopoML markup language document that conforms to the TopoML document type definition represented in the "TopoML.dtd" document. The date of the "TopoML.dtd" document is redacted, but the true date is long prior to April 30, 2001. This document is hereby submitted as probative of a reduction to practice of the subject invention prior to April 30, 2001.

9. Attached, as Exhibit 3, is a true and correct redacted copy of the contents of a document entitled "TopoML.xsd", which is referred to as "W3CSchema" in the document of Exhibit 1. This document is an XML Schema Definition (XSD) which, similar the "TopoML.dtd" document of Exhibit 2, describes the elements in a "TopoML" document and represents the interrelationship between the attributes and elements of a "TopoML" document. The CGI program of the TopoML application (referred to in Section 6 of this affidavit) operates to retrieve from a data source and to convert topology information to a TopoML markup language document that conforms to the TopoML schema represented in the "TopoML.xsd" document. The date of the "TopoML.xsd"

document is redacted, but the true date is long prior to April 30, 2001. This document is hereby submitted as probative of a reduction to practice of the subject invention prior to April 30, 2001.

10. Attached, as Exhibit 4, is a true and correct redacted copy of the contents of a document entitled "Hierarchy For All Packages". This document contains a list of classes and a list of interfaces associated with the TopoML language and application. The applet of the TopoML application (referred to in Section 6 of this affidavit) was implemented based on the classes and interfaces listed in the "Hierarchy For All Packages" document. The date of this document is redacted, but the true date is long prior to April 30, 2001. This document is hereby submitted as probative of a reduction to practice of the subject invention prior to April 30, 2001.

11. Attached, as Exhibit 5, is a true and correct redacted copy of a portion of the contents of a document entitled "Index". This document contains an alphabetic list of classes, interfaces, constructors, methods and fields associated with "TopoML". The applet of the TopoML application (referred to in Section 6 of this affidavit) was implemented utilizing at least some of the classes, interfaces, constructors, methods and fields described in the "Index" document. The date of this document is redacted, but the true date is long prior to April 30, 2001. This document is hereby submitted as probative of a reduction to practice of the subject invention prior to April 30, 2001.

12. Attached, as Exhibit 6, is a true and correct redacted copy of the contents of an electronic mail from Cassio Goldschmidt, providing the status of the "AD topology application" (an implementation of the TopoML application) project. The date of this document is redacted, but the true date is long prior to April 30, 2001. This document indicates that the application build was completed over the TopoML language constructs, and that the application was executable from a browser. This document further indicates that a demonstration of the functionality of the application was performed prior to the date of this document. This e-mail document is maintained on a server operated by Cisco Systems, Inc., and is hereby submitted as probative of a reduction to practice of the subject invention prior to April 30, 2001.

13. Attached, as Exhibit 7, is a true and correct redacted copy of the contents of an electronic mail from Cassio Goldschmidt, inviting internal developer colleagues to

assist in testing the "AD application" (an implementation of the TopoML application). The date of this document is redacted, but the true date is long prior to April 30, 2001. This document indicates that the application was "currently up and running" on a production server ("EMAN-DEV") operated by Cisco Systems, Inc. This e-mail document is maintained on a server operated by Cisco Systems, Inc., and is hereby submitted as probative of a reduction to practice of the subject invention prior to April 30, 2001.

14. All of the acts set forth herein occurred or were performed by me in the United States.

15. At all times alleged herein, I carried out all of the acts set forth herein in secret and internal to my employer, Cisco Systems, Inc. To the best of my knowledge, any substantive disclosures to any parties external to my employer were protected from disclosure by non-disclosure agreements.

16. All documents and acts set forth herein in this Declaration relate to internal software development. None of the acts set forth herein involve public use, sale, offer for sale, publication, or other unprotected disclosure of the subject invention.

Each person signing below hereby declares that all statements made herein of her/his own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Executed at LOS ANGELES, CALIFORNIA on the date set forth below.
(city) (state)

Cassio B. Goldschmidt

Dated: 3/22/2005

Exhibit 1

Appl. No. 09/905,306
Docket No. 50325-0552 (Seq. No. 4434)

TopoML

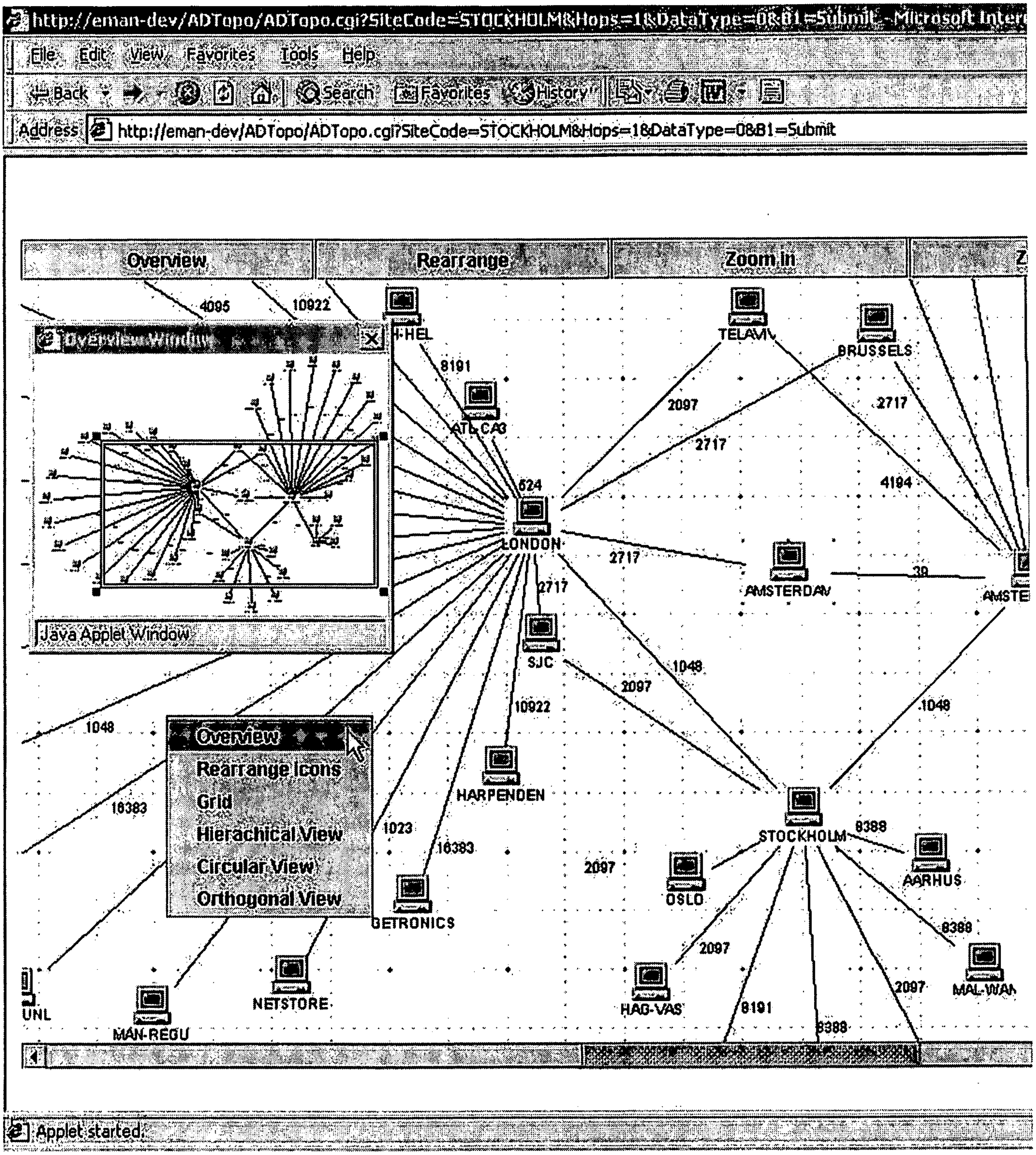
Purpose

TopoML is a XML derived language that was designed to plot graphs (nodes and edges). All you need to do is to pass the nodes and how they connect to each other (edges) and TopoML will take care of doing the best layout for you automatically.

Further, TopoML can do incremental layouts and topologies. For instance, when plotting a very complex network topology, you may want to show the content of some nodes only when the user double clicks it. The TopoML viewer will then go back to the server, get a TopoML description of what's supposed to do and execute it.

TopoML is not only restricted to plotting graphs, it can also display webpages, make use of popup menus, toolbars and much more.

Here is a screenshot of an application that uses TopoML



What's needed to use TopoML

TopoML is implemented as a Java Applet and therefore can be used as a standalone application running in a client machine inside the appletviewer or inside a web browser that supports java such as IE or Netscape. TopoML makes use of Java swing library and in order to have it running inside a browser you must download SUN's HTML converter and convert your HTML file. This conversion will make your HTML able to automatically download the right swing plug-in in case a client that does not have it install tries to see your webpage.

The TopoML language

The language is all implemented using both XML ***and*** Applets parameters. In other words, you can use all the functionality of the language without writing a single line of XML. Look at the HTML source code of this sample to check how to use TopoML as Applet parameters. The same sample can also generate XML and might be useful to look at.

As mentinoned earlier, TopoML works in a incremental way. Let's say you have node A and B in the screen and now you want to add node C connection to B. All you need to do is send node C and specify the connection. Since node B is already in the screen you don't need to re-specify it. In case you do, nothing bad will happen, in fact the the specification will be simply ignored, unless you are changing one of the nodes attributes such as icons, popup menus, tooltips and etc.

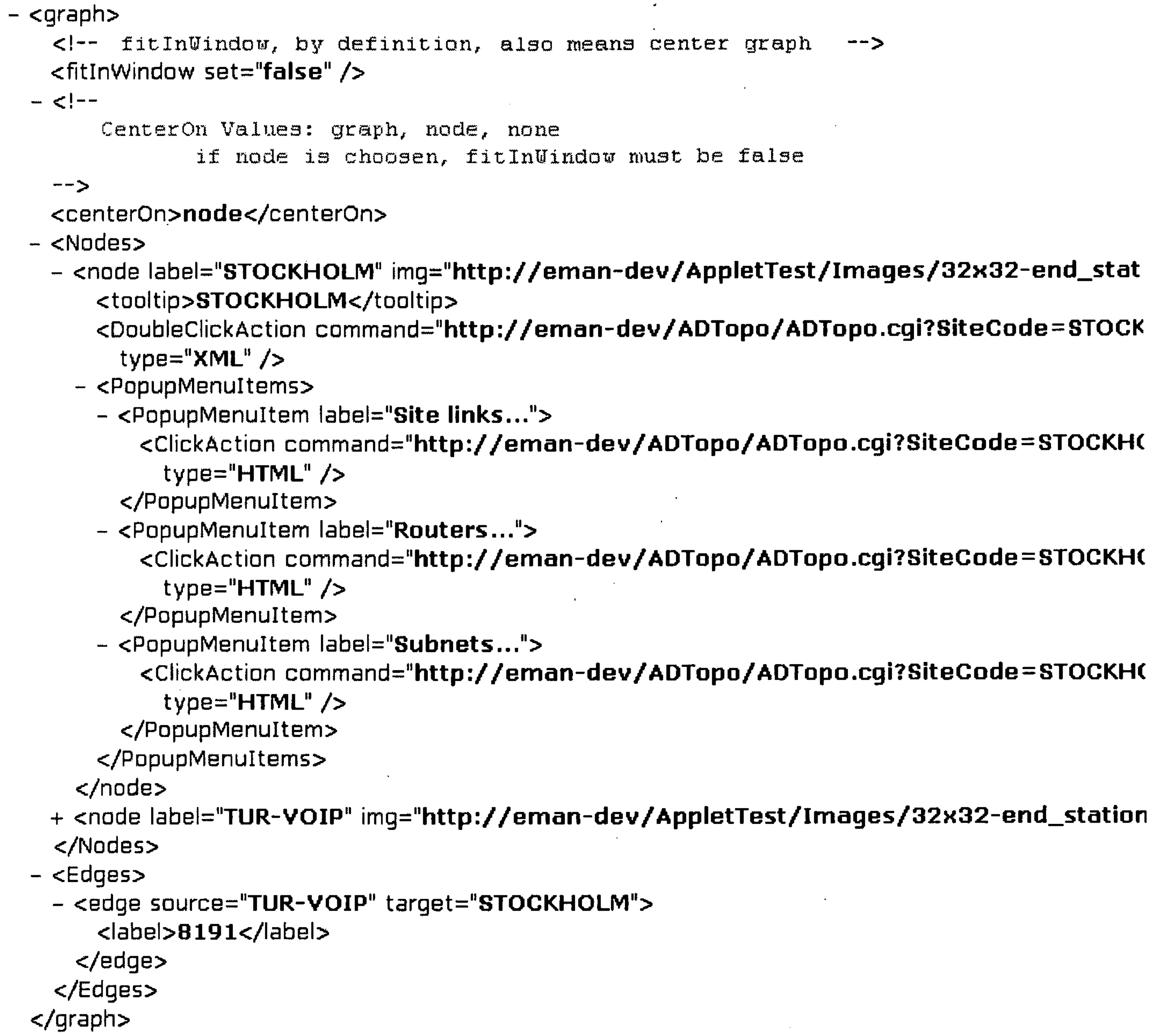
Also note that everything that is once specified remains the same until you declare it differently. For example, if you specify the layout to be Circular and after you add a couple more nodes, the viewer will try to accomodate the new nodes using a circular layout. In case you want to change it to something else, you must specify it.

TopoML comes with several defaults for all it's properties. For example, the default for the grid option is off. Look at the TopoML DTD for the all available tags, their nesting and the defaults.

Elements of the Language

The entire description of TopoML elements can be found in this pdf file that visually explains the language. From the same site you can get the TopoML DTD and W3CSchema. All of this links are full of comments and should provide enought explanation of all possible tags of TopoML.

Example of TopoML on XML
In this example STOCKHOLM connects to TUR-VOIP, note that TUR-VOIP appear contracted here just for the sake of space purpose.



```
- <graph>
    <!--  fitInWindow, by definition, also means center graph   -->
    <fitInWindow set="false" />
  - <!--
        CenterOn Values: graph, node, none
                if node is choosen, fitInWindow must be false
    -->
    <centerOn>node</centerOn>
  - <Nodes>
    - <node label="STOCKHOLM" img="http://eman-dev/AppletTest/Images/32x32-end_stat
        <tooltip>STOCKHOLM</tooltip>
        <DoubleClickAction command="http://eman-dev/ADTopo/ADTopo.cgi?SiteCode=STOCK
          type="XML" />
      - <PopupMenuItems>
        - <PopupMenuItem label="Site links...">
            <ClickAction command="http://eman-dev/ADTopo/ADTopo.cgi?SiteCode=STOCKH
              type="HTML" />
          </PopupMenuItem>
        - <PopupMenuItem label="Routers...">
            <ClickAction command="http://eman-dev/ADTopo/ADTopo.cgi?SiteCode=STOCKH
              type="HTML" />
          </PopupMenuItem>
        - <PopupMenuItem label="Subnets...">
            <ClickAction command="http://eman-dev/ADTopo/ADTopo.cgi?SiteCode=STOCKH
              type="HTML" />
          </PopupMenuItem>
        </PopupMenuItems>
      </node>
    + <node label="TUR-VOIP" img="http://eman-dev/AppletTest/Images/32x32-end_station
    </Nodes>
  - <Edges>
    - <edge source="TUR-VOIP" target="STOCKHOLM">
        <label>8191</label>
      </edge>
    </Edges>
  </graph>
```

...and here is the same data using applet Parameters instead of XML.

```
<PARAM NAME = "nodeImage0" VALUE ="http://eman-dev/AppletTest/Images/32x32-end_station_blue.gif">
<PARAM NAME = "nodeLabel0" VALUE ="STOCKHOLM">
<PARAM NAME = "nodeTooltip0" VALUE ="STOCKHOLM">
<PARAM NAME = "NodeDoubleClickActionCommand0" VALUE ="http://eman-dev/ADTopo/ADTopo.cgi?
SiteCode=STOCKHOLM&Hops=1&DataType=1">
<PARAM NAME = "NodeDoubleClickActionType0" VALUE ="XML">
<PARAM NAME = "NodeTotalPopupMenuItems0" VALUE ="3">
<PARAM NAME = "NodePopupMenuItem0Label0" VALUE ="Site links...">
<PARAM NAME = "NodePopupMenuItem0ClickActionCommand0" VALUE ="http://eman-dev/ADTopo/ADTopo.cgi?
SiteCode=STOCKHOLM&Hops=1&DataType=3">
<PARAM NAME = "NodePopupMenuItem0ClickActionType0" VALUE ="HTML">
<PARAM NAME = "NodePopupMenuItem0Label1" VALUE ="Routers...">
<PARAM NAME = "NodePopupMenuItem0ClickActionCommand1" VALUE ="http://eman-dev/ADTopo/ADTopo.cgi?
SiteCode=STOCKHOLM&Hops=1&DataType=2">
<PARAM NAME = "NodePopupMenuItem0ClickActionType1" VALUE ="HTML">
<PARAM NAME = "NodePopupMenuItem0Label2" VALUE ="Subnets...">
```

```
<PARAM NAME = "NodePopupMenuItem0ClickActionCommand2" VALUE ="http://eman-dev/ADTopo/ADTopo.cgi?
SiteCode=STOCKHOLM&Hops=1&DataType=4">
<PARAM NAME = "NodePopupMenuItem0ClickActionType2" VALUE ="HTML">

<PARAM NAME = "nodeImage1" VALUE ="http://eman-dev/AppletTest/Images/32x32-end_station_blue.gif">
<PARAM NAME = "nodeLabel1" VALUE ="TUR-VOIP">
<PARAM NAME = "nodeTooltip1" VALUE ="TUR-VOIP">
<PARAM NAME = "NodeDoubleClickActionCommand1" VALUE ="http://eman-dev/ADTopo/ADTopo.cgi?SiteCode=TUR-
VOIP&Hops=1&DataType=1">
<PARAM NAME = "NodeDoubleClickActionType1" VALUE ="XML">
<PARAM NAME = "NodeTotalPopupMenuItems1" VALUE ="3">
<PARAM NAME = "NodePopupMenuItem1Label0" VALUE ="Site links...">
<PARAM NAME = "NodePopupMenuItem1ClickActionCommand0" VALUE ="http://eman-dev/ADTopo/ADTopo.cgi?SiteCode=TUR-
VOIP&Hops=1&DataType=3">
<PARAM NAME = "NodePopupMenuItem1ClickActionType0" VALUE ="HTML">
<PARAM NAME = "NodePopupMenuItem1Label1" VALUE ="Routers...">
<PARAM NAME = "NodePopupMenuItem1ClickActionCommand1" VALUE ="http://eman-dev/ADTopo/ADTopo.cgi?SiteCode=TUR-
VOIP&Hops=1&DataType=2">
<PARAM NAME = "NodePopupMenuItem1ClickActionType1" VALUE ="HTML">
<PARAM NAME = "NodePopupMenuItem1Label2" VALUE ="Subnets...">
<PARAM NAME = "NodePopupMenuItem1ClickActionCommand2" VALUE ="http://eman-dev/ADTopo/ADTopo.cgi?SiteCode=TUR-
VOIP&Hops=1&DataType=4">
<PARAM NAME = "NodePopupMenuItem1ClickActionType2" VALUE ="HTML">

<PARAM NAME = "edgeSource0" VALUE ="TUR-VOIP">
<PARAM NAME = "edgeTarget0" VALUE ="STOCKHOLM">
<PARAM NAME = "edgeLabel0" VALUE ="8191">
```

Knowing what's going on

To know what goes on with TopoML while running on a webbrowser you can enable the Java console window once you
install the swing plug-in. To do that, go to control panel and double click "Java plug-in" check the option "Show Java console and the start your TopoML application. This option can be unchecked at anytime later.

Exhibit 2

Appl. No. 09/905,306
Docket No. 50325-0552 (Seq. No. 4434)

XML

= version	1.0
= encoding	UTF-8

Comment — edited with XML Spy v3.5 NT (http://www.xmlspy.com) by Cassio Goldschmidt (Cisco Systems)

PopupMenuItems *1 or more sequence of*

PopupMenuItem	1 or more

PopupMenuItem *1 or more sequence of*

ClickAction	1 or more

ClickAction — *EMPTY*

ClickAction *attribute list*

	Name	Type	Values	Presence	Default
1	command	CDATA		#REQUIRED	
2	type	Choice	**Values** Text 1 HTML 2 XML 3 TOPO	#REQUIRED	

DoubleClickActi... — *EMPTY*

DoubleClickAction *attribute list*

	Name	Type	Values	Presence	Default
1	command	CDATA		#REQUIRED	
2	type	Choice	**Values** Text 1 HTML 2 XML 3 TOPO	#REQUIRED	

Comment — useGrid: deault is false

useGrid — *EMPTY*

Comment — fitInWindow (default is true)

fitInWindow — *EMPTY*

Comment — background: default is white without pictures

background — *EMPTY*

Comment

Comment — This is the main element - the root of everythign else

graph *0 or more choice of*

useGrid	0 or 1
fitInWindow	0 or 1
centerOn	0 or 1
background	0 or 1
buttons	0 or 1
layoutType	0 or 1
PopupMenuIte...	0 or 1
nodes	0 or 1
edges	0 or 1

useGrid *attribute list*

	Name	Type	Values	Presence	Default
1	set	Choice	**Values** Text 1 true 2 false	#REQUIRED	

Comment — fitInWindow, by definition, also means center graph

fitInWindow *attribute list*

	Name	Type	Values	Presence	Default
1	set	Choice	**Values** Text 1 true 2 false	#REQUIRED	

Comment — CenterOn Values: graph, node, none if node is choosen, fitInWindow must be false (default is none)

centerOn — *#PCDATA*

Comment — Img ="null" means clean-up current img, otherwise use Img= a URL. color is a color entry in Hexadecimal (e.g. FFFFFF for white, FF0000 for red). center is a comma separated x,y coordinate.

background *attribute list*

	Name	Type	Values	Presence	Default
1	img	CDATA		#REQUIRED	
2	color	CDATA		#IMPLIED	
3	center	CDATA		#IMPLIED	

Comment — layoutType can be Circular, Hierachical, Orthogonal and ... (default is none)

layoutType — *#PCDATA*

Comment

Comment — Definition of node and nodes

nodes *1 or more sequence of*

node

node *0 or more choice of*

tooltip	0 or 1
location	0 or 1
DoubleClickActi...	0 or 1
PopupMenuIte...	0 or 1

node *attribute list*

	Name	Type	Values	Presence	Default
1	label	CDATA		#REQUIRED	
2	img	CDATA		#REQUIRED	

Comment

Comment — Definition of toolbar buttons

buttons *1 or more sequence of*

button

button *1 or more sequence of*

tooltip	0 or 1
ClickAction	1 or more

		Elm **tooltip**	*0 or 1*			
button *attribute list*						
		Att *Name*	Att *Type*	Att *Values*	Att *Presence*	Att *Default*
	1	label	*CDATA*		*#REQUIRED*	
<!-- *Comment*		A tooltip for the node/button				
Elm **tooltip**		*#PCDATA*				
<!-- *Comment*		The x,y location for a node				
Elm **location**		*#PCDATA*				
<!-- *Comment*						
<!-- *Comment*		Definition of edges and edge				
edges *1 or more sequence of*						
		Elm **edge**	*1 or more*			
edge *0 or more choice of*						
		Elm **label**	*0 or 1*			
		Elm **color**	*0 or 1*			
		Elm **type**	*0 or 1*			
		Elm **PopupMenuIte...**	*0 or 1*			
edge *attribute list*						
		Att *Name*	Att *Type*	Att *Values*	Att *Presence*	Att *Default*
	1	source	*CDATA*		*#REQUIRED*	
	2	target	*CDATA*		*#REQUIRED*	
<!-- *Comment*		Edge's label, default is none				
Elm **label**		*#PCDATA*				
<!-- *Comment*		Edge's color in Hexadecimal (e.g 000000 is black, 00FF00 is green), default is black				
Elm **color**		*#PCDATA*				
<!-- *Comment*		Type: A number that represents the edge to be used: 1=One way arrowed edge, 2=One way reversed arrowed adge, 3=double arrow edge (both directions), 4=no arrows (default)				
Elm **type**		*#PCDATA*				

Exhibit 3

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 NT (http://www.xmlspy.com) by Cassio Goldschmidt
(Cisco Systems) -->
<!--W3C Schema generated by XML Spy v3.5 NT (http://www.xmlspy.com)-->
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
elementFormDefault="qualified">
        <xsd:element name="ClickAction">
                <xsd:complexType>
                        <xsd:attribute name="command" type="xsd:string"
use="required"/>
                        <xsd:attribute name="type" use="required">
                                <xsd:simpleType>
                                        <xsd:restriction base="xsd:NMTOKEN">
                                                <xsd:enumeration value="HTML"/>
                                                <xsd:enumeration value="XML"/>
                                                <xsd:enumeration value="TOPO"/>
                                        </xsd:restriction>
                                </xsd:simpleType>
                        </xsd:attribute>
                </xsd:complexType>
        </xsd:element>
        <xsd:element name="DoubleClickAction">
                <xsd:complexType>
                        <xsd:attribute name="command" type="xsd:string"
use="required"/>
                        <xsd:attribute name="type" use="required">
                                <xsd:simpleType>
                                        <xsd:restriction base="xsd:NMTOKEN">
                                                <xsd:enumeration value="HTML"/>
                                                <xsd:enumeration value="XML"/>
                                                <xsd:enumeration value="TOPO"/>
                                        </xsd:restriction>
                                </xsd:simpleType>
                        </xsd:attribute>
                </xsd:complexType>
        </xsd:element>
        <xsd:element name="PopupMenuItem">
                <xsd:complexType>
                        <xsd:sequence maxOccurs="unbounded">
                                <xsd:element ref="ClickAction"
maxOccurs="unbounded"/>
                        </xsd:sequence>
                </xsd:complexType>
        </xsd:element>
        <xsd:element name="PopupMenuItems">
                <xsd:complexType>
                        <xsd:sequence maxOccurs="unbounded">
                                <xsd:element ref="PopupMenuItem"
maxOccurs="unbounded"/>
                        </xsd:sequence>
                </xsd:complexType>
        </xsd:element>
        <xsd:element name="background">
                <xsd:complexType>
                        <xsd:attribute name="img" type="xsd:string" use="required"/>
                        <xsd:attribute name="color" type="xsd:string"/>
                        <xsd:attribute name="center" type="xsd:string"/>
                </xsd:complexType>
        </xsd:element>
        <xsd:element name="button">
                <xsd:complexType>
                        <xsd:sequence maxOccurs="unbounded">
                                <xsd:element ref="tooltip" minOccurs="0"/>
                                <xsd:element ref="ClickAction" 
maxOccurs="unbounded"/>
                                <xsd:element ref="tooltip" minOccurs="0"/>
                        </xsd:sequence>
                        <xsd:attribute name="label" type="xsd:string" 
use="required"/>
                </xsd:complexType>
        </xsd:element>
        <xsd:element name="buttons">
                <xsd:complexType>
                        <xsd:sequence maxOccurs="unbounded">
                                <xsd:element ref="button"/>
                        </xsd:sequence>
                </xsd:complexType>
        </xsd:element>
        <xsd:element name="centerOn" type="xsd:string"/>
        <xsd:element name="color" type="xsd:string"/>
        <xsd:element name="edge">
                <xsd:complexType>
                        <xsd:choice minOccurs="0" maxOccurs="unbounded">
                                <xsd:element ref="label" minOccurs="0"/>
                                <xsd:element ref="color" minOccurs="0"/>
                                <xsd:element ref="type" minOccurs="0"/>
                                <xsd:element ref="PopupMenuItems" minOccurs="0"/>
                        </xsd:choice>
                        <xsd:attribute name="source" type="xsd:string" 
use="required"/>
                        <xsd:attribute name="target" type="xsd:string" 
use="required"/>
                </xsd:complexType>
        </xsd:element>
        <xsd:element name="edges">
                <xsd:complexType>
                        <xsd:sequence maxOccurs="unbounded">
                                <xsd:element ref="edge" maxOccurs="unbounded"/>
                        </xsd:sequence>
                </xsd:complexType>
        </xsd:element>
        <xsd:element name="fitInWindow">
                <xsd:complexType>
                        <xsd:attribute name="set" use="required">
                                <xsd:simpleType>
                                        <xsd:restriction base="xsd:NMTOKEN">
                                                <xsd:enumeration value="true"/>
                                                <xsd:enumeration value="false"/>
                                        </xsd:restriction>
                                </xsd:simpleType>
                        </xsd:attribute>
                </xsd:complexType>
        </xsd:element>
        <xsd:element name="graph">
                <xsd:complexType>
                        <xsd:choice minOccurs="0" maxOccurs="unbounded">
                                <xsd:element ref="useGrid" minOccurs="0"/>
                                <xsd:element ref="fitInWindow" minOccurs="0"/>
                                <xsd:element ref="centerOn" minOccurs="0"/>
                                <xsd:element ref="background" minOccurs="0"/>
                                <xsd:element ref="buttons" minOccurs="0"/>
                                <xsd:element ref="layoutType" minOccurs="0"/>
                                <xsd:element ref="PopupMenuItems" minOccurs="0"/>
                                <xsd:element ref="nodes" minOccurs="0"/>
                                <xsd:element ref="edges" minOccurs="0"/>
                        </xsd:choice>
                </xsd:complexType>
        </xsd:element>
        <xsd:element name="label" type="xsd:string"/>
        <xsd:element name="layoutType" type="xsd:string"/>
        <xsd:element name="location" type="xsd:string"/>
        <xsd:element name="node">
                <xsd:complexType>
                        <xsd:choice minOccurs="0" maxOccurs="unbounded">
                                <xsd:element ref="tooltip" minOccurs="0"/>
                                <xsd:element ref="location" minOccurs="0"/>
                                <xsd:element ref="DoubleClickAction" minOccurs="0"/>
                                <xsd:element ref="PopupMenuItems" minOccurs="0"/>
                        </xsd:choice>
                        <xsd:attribute name="label" type="xsd:string"
use="required"/>
                        <xsd:attribute name="img" type="xsd:string" use="required"/>
                </xsd:complexType>
        </xsd:element>
        <xsd:element name="nodes">
                <xsd:complexType>
                        <xsd:sequence maxOccurs="unbounded">
                                <xsd:element ref="node"/>
                        </xsd:sequence>
                </xsd:complexType>
        </xsd:element>
        <xsd:element name="tooltip" type="xsd:string"/>
        <xsd:element name="type" type="xsd:string"/>
        <xsd:element name="useGrid">
                <xsd:complexType>
                        <xsd:attribute name="set" use="required">
                                <xsd:simpleType>
                                        <xsd:restriction base="xsd:NMTOKEN">
                                                <xsd:enumeration value="true"/>
                                                <xsd:enumeration value="false"/>
                                        </xsd:restriction>
                                </xsd:simpleType>
                        </xsd:attribute>
                </xsd:complexType>
        </xsd:element>
</xsd:schema>
```

Exhibit 4

Overview Package Class Use **Tree** **Index** **Help**
PREV NEXT **FRAMES** **NO FRAMES**

Hierarchy For All Packages

Package Hierarchies:
<default>

Class Hierarchy

- class **CActionObject**
- class **CConvUtil**
- class **CGraphDoc**
- class **CGraphView**
- class **CMyButton**
- class **CMyEdge**
- class **CMyMenuItem**
- class **CMyNodeEx**
- class **CMyPopupMenu**
- class **CParamEnvironment** (implements .IEnvironment)
- class **CParamParse**
- class **CParamTopoEdge** (implements .ITopoEdge)
- class **CParamTopoNode** (implements .ITopoNode)
- class **CRes**
- class **CXMLEnvironment** (implements .IEnvironment)
- class **CXMLHelper**
- class **CXMLTopoEdge** (implements .ITopoEdge)
- class **CXMLTopoNode** (implements .ITopoNode)
- class java.lang.Object
    - class java.awt.Component
        - class java.awt.Container
            - class javax.swing.JComponent
                - class tomsawyer.editor.TSEGraphWindow
                    - class **ExGraphWindow**
            - class java.awt.Panel
                - class java.applet.Applet
                    - class javax.swing.JApplet
                        - class **Applet1** (implements ActionListener)
            - class java.awt.Window
                - class java.awt.Dialog
                    - class javax.swing.JDialog
                        - class **CWebBrowserDlg** (implements HyperlinkListener)
    - class java.util.ResourceBundle
        - class java.util.ListResourceBundle
            - class **ExResourceBundle**
    - class tomsawyer.editor.TSEWindowState
        - class tomsawyer.editor.TSEWindowInputState

- class tomsawyer.editor.state.TSESelectState
  - class **MyExSelectState** (implements ActionListener)
- class java.awt.event.WindowAdapter
  - class **Applet1.WindowHandler**

Interface Hierarchy

- interface **ITopoElement**
  - interface **IEnvironment**
  - interface **ITopoEdge**
  - interface **ITopoNode**
- interface **TomahawkConstants**

Overview Package Class Use **Tree** **Index** **Help**
PREV NEXT **FRAMES** **NO FRAMES**

Exhibit 5

Appl. No. 09/905,306
Docket No. 50325-0552 (Seq. No. 4434)

Overview Package Class Use **Tree** **Index** **Help**
PREV NEXT **FRAMES** **NO FRAMES**
< A B C D E F G H I J L M N O P R S T U W Z

<

<default> - Class Diagram

A

ACTION_ABORT - Static variable in interface TomahawkConstants
This command string instructs the application to abort any action that it is currently carrying out.
actionPerformed(java.awt.event.ActionEvent) - Method in class Applet1
This class reacts to events posted by menus and toolbars.
actionPerformed(java.awt.event.ActionEvent) - Method in class MyExSelectState
This method responds to the actions caused by the user selecting one of the menu items.
ADD_LABEL - Static variable in interface TomahawkConstants
This command string instructs the application to add a label to the currently selected edge.
addColection(IEnvironment,java.lang.String,java.lang.String) - Method in class CGraphDoc
Generic Method to add ALL items to a collection.
addMenuItem(ITopoNode) - Method in class CMyPopupMenu
Reads menu items from the ITopoNode input and insert in this structure.
addNotify() - Method in class CWebBrowserDlg

addPopupMenu(ITopoElement,java.lang.String,java.lang.String) - Method in class CMyPopupMenu
Given the source (where the class should read the menu from), and the name of the collection (sPopupName) and the itens inside the collection (sPopupItemName).
APP_ABOUT - Static variable in interface TomahawkConstants
This command string instructs the application to display the about message box.
APP_EXIT - Static variable in interface TomahawkConstants
This command string instructs the application to terminate.
appButtons - Variable in class ExResourceBundle
This variable stores all buttons created by this application.
Applet1 - class Applet1
This is the main class that creates and drives the execution of the applet, the root of all evil...
Applet1.WindowHandler - class Applet1.WindowHandler
This class implements a window handler.
Applet1() - Constructor for class Applet1

APPLY_LAYOUT - Static variable in interface TomahawkConstants
This command string instructs the application to apply a specified layout to the graph.

A wrapper around JMenuItem to execute Topo commands
CMyMenuItem(java.lang.String,CActionObject) - Constructor for class CMyMenuItem

CMyNodeEx - class CMyNodeEx
This class is basically a wrapper around Tom Sawyer's TSENode class with some extra functionality added such as the ability to execute commands and tooltips.
CMyNodeEx(tomsawyer.editor.TSENode,java.lang.String,CActionObject) - Constructor for class CMyNodeEx

CMyPopupMenu - class CMyPopupMenu
My popup menu class wrapper.
CMyPopupMenu() - Constructor for class CMyPopupMenu
Constructor
COPY_GRAPH - Static variable in class CRes

COPY_GRAPH - Static variable in interface TomahawkConstants
This command string instructs the application to copy the selected objects to the clipboard.
CParamEnvironment - class CParamEnvironment
Implementation of the Environment interface using Applet parameters to read all the information from.
CParamEnvironment() - Constructor for class CParamEnvironment

CParamParse - class CParamParse
Helper class that concentrates all the methods that several classes use to get information from the Applet parameter.
CParamParse() - Constructor for class CParamParse

CParamTopoEdge - class CParamTopoEdge
Implementation of the Edge interface using Applet parameter to read all the information from.
CParamTopoEdge(int,java.lang.String) - Constructor for class CParamTopoEdge

CParamTopoNode - class CParamTopoNode
Title: Topology Applet Description: An applet to generate generic/network topology (graphs).
CParamTopoNode(int,java.lang.String) - Constructor for class CParamTopoNode

CREATE_CHILD_GRAPH - Static variable in interface TomahawkConstants
This command string instructs the application to create a child graph and associate it with the node hit by the mouse cursor.
CREATE_EDGE_STATE - Static variable in interface TomahawkConstants
This command string instructs the application to switch to the edge creation mode.
CREATE_NODE_STATE - Static variable in interface TomahawkConstants
This command string instructs the application to switch to the node creation mode.
createMenuBar(java.awt.event.ActionListener) - Method in class ExResourceBundle
This method generates a menu bar from the resource.
createPopup(java.lang.String,java.awt.event.ActionListener) - Method in class ExResourceBundle
This method creates a named popup menu.
CreatePopupMenu(ITopoElement) - Method in class CMyEdge

CreatePopupMenu(ITopoElement) - Method in class CMyNodeEx
Reads data from the environment and creates a popup menu
CreatePopupMenu(ITopoNode) - Method in class CMyEdge

createToolBar(java.lang.String,java.awt.event.ActionListener) - Method in class ExResourceBundle
This method creates a toolbar specified by a given name in the resource table.
CRes - class CRes

CUT_GRAPH - Static variable in interface TomahawkConstants
This command string instructs the application to cut the selected objects from the graph and place them in the clipboard.
CWebBrowserDlg - class CWebBrowserDlg

CWebBrowserDlg() - Constructor for class CWebBrowserDlg

CWebBrowserDlg(java.awt.Frame) - Constructor for class CWebBrowserDlg

CWebBrowserDlg(java.lang.String) - Constructor for class CWebBrowserDlg

CXMLEnvironment - class CXMLEnvironment
Implementation of the Environment interface using XML to read all the information from.
CXMLEnvironment(java.lang.String) - Constructor for class CXMLEnvironment

CXMLHelper - class CXMLHelper
Helper class that concentrates all the XML methods that several classes use to get information from the XML data source.
CXMLHelper(org.w3c.dom.Node) - Constructor for class CXMLHelper

CXMLTopoEdge - class CXMLTopoEdge
Implementation of the Edge interface using XML to read all the information from.
CXMLTopoEdge(org.w3c.dom.Node) - Constructor for class CXMLTopoEdge

CXMLTopoNode - class CXMLTopoNode
Implementation of the Node interface using XML to read all the information from.
CXMLTopoNode(org.w3c.dom.Node) - Constructor for class CXMLTopoNode

---

D

DEFAULT_EDGE_UI - Static variable in interface TomahawkConstants
This constant defines the name for the default edge UI type.
DEFAULT_NODE_UI - Static variable in interface TomahawkConstants
This constant defines the name for the default node UI type.
DELETE_BACKGROUND_IMAGE - Static variable in interface TomahawkConstants
This command string instructs the application to delete the background image of the current graph.
DELETE_CHILD_GRAPH - Static variable in interface TomahawkConstants
This command string instructs the application to delete the child graph of the currently selected node.
DELETE_SELECTED - Static variable in interface TomahawkConstants
This command string instructs the application to delete the selected objects from the graph.

DUPLICATE_GRAPH - Static variable in interface TomahawkConstants
This command string instructs the application to create a copy of all selected objects in the graph.

E

EDGE_ARROW - Static variable in interface TomahawkConstants
This command string instructs the application to change the arrow type on selected edges.
EDGE_POPUP - Static variable in interface TomahawkConstants
This constant defines the command string associated with an edge popup menu.
EDIT_BACKGROUND - Static variable in interface TomahawkConstants
This command string instructs the application to edit the background color of the selected objects.
EDIT_BORDER - Static variable in interface TomahawkConstants
This command string instructs the application to edit the border color of the selected objects.
EDIT_FONT - Static variable in interface TomahawkConstants
This command string instructs the application to edit the font of the selected objects.
EDIT_FOREGROUND - Static variable in interface TomahawkConstants
This command string instructs the application to edit the foreground color of the selected objects.
EDIT_TEXT - Static variable in interface TomahawkConstants
This command string instructs the application to edit the text (tag) of the selected objects.
execute() - Method in class CMyButton
Executes an action when the toolbar is clicked
execute() - Method in class CMyNodeEx
Executes a topo command.
execute(CMyNodeEx) - Method in class CActionObject
Executes the command related to a certain node.
execute(CMyNodeEx) - Method in class CMyMenuItem
Executes a topo command
ExGraphWindow - class ExGraphWindow
This class extends the main toolkit class TSEGraphWindow in order to provide tooltip text over the graph objects.
ExGraphWindow(tomsawyer.editor.TSEGraph,boolean) - Constructor for class ExGraphWindow

ExResourceBundle - class ExResourceBundle
This class defines a resource bundle that has the ability to build various UI objects directly from resources.

F

fitInWindow() - Method in class CGraphView
Fits the entire graph in the view (applet window)
frameSizeAdjusted - Variable in class CWebBrowserDlg

Exhibit 6

[redacted]

-----Original Message-----
From: Goldschmidt, Cassio [mailto:cgoldsch@cisco.com]

Subject: Topology Status

Ray,

Here is the status of the topology project: I wake up 6:30AM and you can page me if you have any questions.

Cassio

Currently we have the AD topology application completed build over my TOPOML language and running on a browser. I showed to both Robbie and Alan today and both seemed to be very pleased with the results so far and personally I feel the same as TOPOML seems to be very powerful and can be reused for many other applications on EMAN and beyond.

Exhibit 7

-----Original Message-----
From: Goldschmidt, Cassio [mailto:cgoldsch@cisco.com]

To: Eman-Topo-Dev@Cisco. Com

Subject: Let's break Topo down!

Feeling angry today?

Please help me test topo. The AD application is currently up and running on EMAN-DEV and I need to load test it as well as to get ideas and bug reports. Fortunately testing this app is really fun! :-)

http://eman-dev/ADTopo/TestPage.htm

If you press submit with the data you already have filled in the form you will get the topology for STOCKHOLM. Of course you can change that freely however at this point I'll ask you to only input a site code that does exists as when we integrate this part if Robbie's there will be no free typing for names and therefore no need to worry about that.

For the ones viewing topo for the very first time, it will ask you to get the swing plug-in from Sun Microsystems. The app will then install the plug-in and start topo. This obviously only happens the first time you run it.

Good news from yesterday to today:
- Topo is running on EMAN-DEV (a Unix box) and not my holly laptop (big thanks to Josh).
- The back browser button does not confuses TOPO however the topology goes back to the initial state.
- Applet size is set to give optimal view for a user on 1024x768 using a browser on full screen.
- No debugging red icons anymore